

AI-02052 Ubuntu Server in Azure

Owner	© Rigel Arcayan
Tags	Project
Created time	@September 1, 2025 10:42 AM

I have an Office 365 business account, I would like to test Ubuntu Server in Azure for testing OpenEMR and Wordpress, guide me step by step from beginning to end. I've already deployed similar setup in my more server. I'm behind Cgnat which I think offloading self-hosted is the solution. This is just for testing and learning purposes and I don't like to incur unnecessary charges on my credit card. Keep this project simple, efficient, cost effective and internet facing Wordpress and OpenEMR.

Since you have an existing Office 365 business account, you can use it to sign in to the Azure Portal. You'll want to leverage Azure's free tier services to keep costs down. You can deploy a single, small Ubuntu Server virtual machine (VM) and install a LAMP (Linux, Apache, MySQL, PHP) or LEMP (Linux, Nginx, MySQL, PHP) stack to host both applications. This approach is more cost-effective than deploying separate VMs for each.

Here's a step-by-step guide to get you set up. 🚀

1. Planning and Cost Management

- **Azure Free Account:** Your Office 365 business account does not automatically grant you access to the Azure Free Account, which is for new customers. However, Azure offers certain services that are always free, like a limited-use Linux VM (B1S size) and a certain amount of storage and outbound data transfer. You'll need to monitor your usage closely to stay within these limits and avoid charges.
- **Virtual Machine Size:** For testing purposes, choose the smallest, most cost-effective VM size available, such as a B1s or B1ms (burstable) VM. These are designed for light workloads and are perfect for a learning environment.

- Pricing Calculator: Before you deploy, use the Azure Pricing Calculator to estimate your monthly costs. This helps you understand what you're paying for and ensures you don't incur surprise charges.
- Resource Management: Create a new resource group specifically for this project. This makes it easy to delete all related resources at once when you're done, preventing any lingering charges.

1. Deploying the Ubuntu Server VM

- Log in to the Azure Portal: Go to the Azure Portal and sign in with your Office 365 business credentials.
- Create a Virtual Machine: Click "Create a resource", search for "Virtual machine", and select "Create".
 - Basics Tab:
 - Subscription: Select your Azure subscription.
 - Resource Group: Click "Create new" and give it a descriptive name like openemr-wordpress-test.
 - Virtual machine name: Give your VM a unique name, e.g., test-server-ubuntu.
 - Region: Choose a region close to your physical location to reduce latency.
 - Availability options: Leave as default.
 - Security type: Leave as default.
 - Image: Select "Ubuntu Server 22.04 LTS" or the latest LTS version.
 - VM architecture: Select "x64".
 - Size: Choose a low-cost VM size like "Standard B1s".
 - Administrator account: Select "SSH public key" for better security. Provide a username (e.g., azureuser) and generate a new key pair. Download your private key; you'll need it to connect. Alternatively, you can use a password if you prefer, but it's less secure.

- Inbound port rules: Select "Allow selected ports". In the dropdown, choose "HTTP (80)", "HTTPS (443)", and "SSH (22)". This will expose the server to the internet for your testing.
- Disks Tab: Leave the defaults for a simple setup. Use the standard OS disk.
- Networking Tab: Ensure the public IP is configured. You'll need this to access your sites.
- Review + create: Review your settings, then click "Create". The deployment will take a few minutes.

Setting Up the Web Server and Databases

- Connect to your VM: Once the VM is deployed, find its public IP address on the VM's overview page. Use an SSH client (like PuTTY on Windows or a terminal on Linux/macOS) to connect using the private key you downloaded. The command will be similar to `ssh -i /path/to/your/private_key azureuser@<public_ip_address>`.
- Update the Server: Once connected, update the package list and upgrade existing packages.


```
sudo apt update
sudo apt upgrade -y
```

Install the LAMP Stack: Install Apache, MySQL, and PHP along with the necessary extensions.

```
sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql php-gd php-curl php-intl php-mbstring php-xml php-zip -y
```

- Secure MySQL: Run the security script to set a strong root password and secure your MySQL installation.


```
sudo mysql_secure_installation
```

Configure Apache Virtual Hosts: Create separate virtual hosts for WordPress and OpenEMR.

- Create directories for each site:


```
sudo mkdir -p /var/www/wordpress.test
sudo mkdir -p /var/www/openemr.test
```

- Create the Apache configuration files for each.
 - For WordPress: `sudo nano /etc/apache2/sites-available/wordpress.conf`
 - For OpenEMR: `sudo nano /etc/apache2/sites-available/openemr.conf`
- In each file, configure the DocumentRoot and ServerName.
- Enable the new sites and the rewrite module, then restart Apache.


```
sudo a2ensite wordpress.conf
sudo a2ensite openemr.conf
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Installing WordPress and OpenEMR

- Create Databases and Users: Log in to MySQL as the root user you configured and create a separate database and user for each application.


```
CREATE DATABASE wordpress_db;
CREATE USER 'wordpress_user'@'localhost' IDENTIFIED BY
'your_strong_password';
GRANT ALL PRIVILEGES ON wordpress_db.* TO
'wordpress_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Repeat this for OpenEMR.
- Install WordPress:
 - Download the latest WordPress archive.
 - Extract it to your WordPress directory and set the correct file permissions.
 - Configure the wp-config.php file with the database credentials.
- Install OpenEMR:
 - Download the OpenEMR archive and extract it to your OpenEMR directory.
 - Set the correct file permissions.
 - Navigate to `http://<public_ip_address>/openemr` in your browser to run the web-based installer and configure the database connection.

1. Final Steps and Cleanup

- Access the sites: You can now access your sites using the public IP address in your browser. Since you haven't configured DNS, you'll need to use the public IP followed by the directory name (e.g., `http://<public_ip>/wordpress` and `http://<public_ip>/openemr`).
- Cost Monitoring: Regularly check the "Cost Management" section in your Azure Portal to monitor your spending and ensure you're staying within your budget.
- Cleanup: When you're finished testing, go to the "Resource groups" page in the Azure Portal, select the `openemr-wordpress-test` resource group, and click "Delete resource group". This will completely remove the VM and all associated resources, preventing any further costs.

How to run Cockpit and Webmin in Ubuntu Azure? Provide a step by step guide

Here's a step-by-step guide on how to install and run Cockpit and Webmin on an Ubuntu virtual machine in Azure. While the process is generally the same as on any Ubuntu server, you'll need to pay attention to a key step regarding firewall rules in Azure.

General Prerequisites

Before you begin, you should have a running Ubuntu virtual machine (VM) in your Azure environment. You will need to connect to this VM via SSH. Ensure the user you are using for the installation has sudo privileges.

Installing Cockpit

Cockpit is a web-based server management tool that is available in Ubuntu's default repositories.

- Connect to your Ubuntu VM
Open your terminal or an SSH client and connect to your Azure VM using its public IP address.
- Install Cockpit
Update your system's package list and then install Cockpit using the following commands:

```
sudo apt update
sudo apt install -y cockpit
```

This command installs the Cockpit package and its dependencies.

- Enable and Start the Cockpit Service

The service should start automatically after installation. To ensure it starts on boot and to check its status, run these commands:

```
sudo systemctl enable --now cockpit.socket
sudo systemctl status cockpit
```

The status output should show the service as active (running).

- Open the Firewall Port in Azure

This is a crucial step for Azure. By default, Azure's network security groups (NSGs) block most incoming traffic. To access Cockpit's web interface, you must allow incoming traffic on port 9090.

- Navigate to your VM in the Azure portal.
- Go to Networking under the Settings section.
- Click Add inbound port rule.
- Set the Destination port ranges to 9090.
- Set the Protocol to TCP.
- Set the Action to Allow.
- Give the rule a name and description, then click Add.

- Access the Cockpit Web Interface

Open a web browser and navigate to

https://<your_VM_public_IP_address>:9090. Replace

<your_VM_public_IP_address> with the public IP of your Azure VM. You may see a security warning about a self-signed certificate, which you can safely bypass. Log in with your Ubuntu user credentials.

Installing Webmin

Webmin is another popular web-based control panel for Linux system administration. It's not in the default repositories, so the installation process is a bit different.

- Add the Webmin Repository

First, you need to add the Webmin repository to your system's sources list and import the GPG key.

```
sudo apt update
```

```
sudo apt install -y software-properties-common apt-transport-https
```

```
wget -qO- https://download.webmin.com/download/repository/webmin-
```

```
keys.asc | sudo apt-key add -
```

```
sudo add-apt-repository "deb
```

```
https://download.webmin.com/download/repository sarge contrib"
```

Alternatively, you can use the official setup script to handle this automatically:

```
curl -o setup-repos.sh
```

```
https://raw.githubusercontent.com/webmin/webmin/master/setup-repos.sh
```

```
sudo sh setup-repos.sh
```

- Install Webmin

Update your package list again and install Webmin:

```
sudo apt update
```

```
sudo apt install -y webmin
```

- Open the Firewall Port in Azure

Similar to Cockpit, you need to allow access to Webmin's default port, which is 10000.

- In the Azure portal, navigate to your VM's Networking settings.
- Click Add inbound port rule.
- Set the Destination port ranges to 10000.
- Set the Protocol to TCP.
- Set the Action to Allow.
- Name the rule and click Add.

- Access the Webmin Web Interface

In a web browser, go to https://<your_VM_public_IP_address>:10000. Log in with your Ubuntu user credentials.

This video provides a quick tutorial on installing and using Cockpit on an Ubuntu server, which is relevant to the process on an Azure VM.

Install Cockpit on Ubuntu 20.04 // Quick introduction to Cockpit The Web-Based Linux Server Manager

YouTube video views will be stored in your YouTube History, and your data will be stored and used by YouTube according to its Terms of Service